

NEKCEM

In the software package NEKCEM (Nekton for Computational ElectroMagnetics) the spectral element discontinuous Galerkin (SEDG) method is employed on a conformal spectral-element mesh. NEKCEM features geometric flexibility, spectrally accurate numerical convergence, and an efficient parallel implementation.

This package consists of the components of meshing, Maxwell time domain solver, electrostatic poisson solver, and visualization tools. It supports the computational domain boundary with perfectly matched layer, perfect conductor, and periodic material.

1. Getting started with NEKCEM

The following sections describe the process for downloading and building the NEKCEM package. It is assumed the user is comfortable with using *Subversion* (<http://subversion.tigris.org/>), as well as with compiling Fortran and C programs.

1.1. Getting the source

NEKCEM is available for download via *the Subversion repository* (<https://trac.mcs.anl.gov/projects/NEKCEM>):

```
svn co https://svn.mcs.anl.gov/repos/NEKCEM
```

Other utilities are available *here* (<http://www-unix.mcs.anl.gov/~mmin/nekcm/nek-helpful-software.tar>). This utilities collection contains the following:

- prex: description goes here...
- postx: description goes here...
- genmap: description goes here...
- n2to3: description goes here...

It is also recommended to download *ParaView* (<http://www.paraview.org/New/index.html>) and *PyVTK* (<http://cens.ioc.ee/projects/pyvtk/>).

1.2. Contents of the NEKCEM package

The NEKCEM package contains the source code, examples (generated by prex and genmap), libraries used, and documentation.

- bin: a collection of scripts for building and running NEKCEM
 - makenek: builds the nekcem package. Run from an 'example' directory via `../bin/makenek`. See `makenek --help` for options
 - nek: runs the nekcem package with 'reasonable' defaults for the platform. See `nek --help` for options
 - arch-settings: contains architecture-specific settings for building NEKCEM. Used by makenek
 - run_paraview: used for visualization. MORE DESCRIPTION GOES HERE
 - post2vtk: convert a .dat dump from FORTRAN into a VTK file for use in paraview
- examples: sample problems generated by prex, genmap
 - tube: DESCRIPTION GOES HERE
 - tube-mw: DESCRIPTION GOES HERE
- libs: BLAS and LAPACK can be placed here if not already installed on your system
- src: source code
- doc: documentation

2. Building and Running NEKCEM

Building and using NEKCEM consist of creating an appropriate mesh and building the package for that mesh

2.1. Mesh generation

1. Invoke the prex GUI tool to generate a mesh from directory where you want the generated files to be stored:

```
cd example/2dnanocyl
../bin/prex
```

2. specify the name to be used for the mesh in prex (i.e. 2dnanocyl)

3. use prex to specify the geometry and boundary conditions of the mesh
4. OTHER INSTRUCTIONS GO HERE

2.2. Compiling NEKCEM

To compile the package, run the makenek script from the directory of the example just created:

```
../../bin/makenek
```

See makenek -h for compile options. The most useful option is probably -d, which builds in debug mode (debug symbols and no optimization).

A few files will get moved around during the compile process:

- The 2dnanocyl usr file will get copied into the ../../src directory as subuser.F.
- The SIZEu file will get copied into the ../../src directory as SIZE.

2.3. Running NEKCEM

To run on the current example, use the nek script

```
../../bin/nek 2dnanocyl
```

See nek -h for options. You'll probably want to use -n for controlling the number of nodes, and -t for the walltime.

A few files will get generated during the execution:

- DESCRIPTION OF LOG FILES GOES HERE
- Field files will be generated in a format 2dnanocyl001.dat in the vtk directory (also created). After conversion with post2vtk these can be viewed in paraview

2.4. Viewing Results

Paraview can be used to view the results for each processor and timestep. The vtk directory contains the files needed by paraview. For more information, see the ParaView website.
(<http://www.paraview.org/New/index.html>)

3. Documentation

The rea-file.txt file located in /doc describes the format of the .rea file generated by prex

Index

D

- database (bibliographic), 253, 255
 - structure, 255
 - tools, 259
- dates (language specific), 179
- DC fonts, 172, 177
 - Math fonts, 177